

KEMI-TORNION AMMATTIKORKEAKOULU

Hankintasovellus

Jouni Möttönen

Tietotekniikan koulutusohjelman opinnäytetyö
Ohjelmistotekniikan suuntautumisvaihtoehto
Insinööri(AMK)

KEMI 2012

ALKUSANAT

Kiitän työn ohjaajaa yliopettaja Tapani Ruokasta saamastani alkusysäyksestä opinnäytetyön aloittamiseen ja PPSHP:n yhteyshenkilöä Matti Pohjolaa kärsivällisyydestä. Kiitoksen ansaitsevat myös Pasi ja Pirkko Tulkki, kun ovat tarjonneet neljän vuoden ajan yösijan viikonloppuisin sekä erilliset kiitokset vielä Pasille, jonka esimerkin innostamana sain idean lähteä näin aikuisiällä vielä opiskelemaan.

Kahden pienen lapsen isänä aikuisopiskelu olisi ollut täysin mahdotonta ilman rakkaan vaimoni tukea ja ymmärrystä, joten suuret kiitokset vaimolleni Tarjalle sekä tyttärellemme Jennille ja pojallemme Santerille.

TIIVISTELMÄ

Kemi-Tornion ammattikorkeakoulu, Tekniikan ala	
Koulutusohjelma	Tietotekniikka
Opinnäytetyön tekijä	Jouni Möttönen
Opinnäytetyön nimi	Hankintasovellus
Työn laji	Opinnäytetyö
päiväys	23.04.2012
sivumäärä	31 + 13 liitesivua
Opinnäytetyön ohjaaja	DI Tapani Ruokanen, yliopettaja
Yritys	PPSHP
Yrityksen yhteyshenkilö/valvoja	Järjestelmäasiantuntija Matti Pohjola

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa selainpohjainen hankintaesitysten käsittelysovellus. Sovelluksen avulla voidaan tehdä hankintaesityksiä, antaa lausuntoja jo tehtyihin esityksiin, tarkistaa aiemmin tehdyn esityksen tilanne ja päivittää hankintaesityksen tilaa. Työn tilaajana toimii Pohjois-Pohjanmaan Sairaanhoidopiiri. Työ sisältää käyttöliittymän, tietokannan ja palvelinohjelmiston suunnittelun ja testiversion toteutuksen.

Ohjelmointityö toteutettiin NetBeans IDE-sovelluskehitysympäristössä. Ohjelmien testausympäristönä toimi Microsoft Virtual PC alustalla toteutettu kopio toimeksiantajan tuotantoympäristöstä. Käyttöliittymä toteutettiin HTML-sivunkuvauskielellä, CSS-tyylinäärityillä ja PHP-ohjelmointikielellä. Sovelluksen toiminnallisuudet on toteutettu PHP- ja JavaScript-ohjelmointikielillä ja ohjelmaan syötetyn tiedon tallennuspaikkana toimii MS SQL-palvelimella sijaitseva relaatiotietokanta.

Johtopäätöksenä työn tuloksesta voitiin todeta paikkansa pitävän työmääräarvion olevan työn onnistumisen kannalta erittäin tärkeässä roolissa. Työn kuluessa tavoitteita jouduttiin karsimaan, koska käytettävissä ollut aika alkoi loppua. Tämä johtui virheellisestä työmääräarviosta.

Asiasanat: PHP, MS SQL, JavaScript, CSS, tietokanta.

ABSTRACT

Kemi-Tornio University of Applied Sciences, Technology	
Degree Programme	Information Technology
Name	Jouni Möttönen
Title	Purchase Order System
Type of Study	Bachelor's Thesis
Date	23 April 2011
Pages	31 + 13 appendices
Instructor	Tapani Ruokanen, M.Sc (Tech.)
Company	PPSHP
Contact Person from Company	Matti Pohjola, System Specialist

The objective of Thesis was to design and implement the web based Purchase Order System. This project was commissioned by Northern Ostrobothnia Hospital District. The Purchase Order System allows a user to do new proposal, give statements to proposal, verify proposal statuses and upgrade proposal statuses. The project included a user interface, database, server application planning and test version implementation.

The software was produced in the software development environment called NetBeans IDE. A user interface was produced with HTML-language and CSS style sheet language. The software core functions were coded using PHP scripting language. MS SQL database was used as the main data store.

In conclusion, planning the estimate workload was one of the most important parts of project. The aim of the project was decreased during the project due to the wrong estimate of the workload.

Keywords: SQL, PHP, CSS, database, JavaScript.

SISÄLLYSLUETTELO

ALKUSANAT	I
TIIVISTELMÄ	II
ABSTRACT	III
SISÄLLYSLUETTELO	IV
KÄYTETYT MERKIT JA LYHENTEET	V
1. JOHDANTO	1
2. TIETOKANNAN SUUNNITTELU	2
2.1. Yleistä tietokannoista	2
2.2. Relaatiotietokannat	3
2.2.1. Relaationtietokantojen osat	3
2.3. SQL	4
2.4. Hankintasovelluksen tietokannan suunnittelu	5
3. WWW-SOVELLUKSEN SUUNNITTELU	9
3.1. Käyttötapaukset	10
3.1.1. Esityksen tekijän käyttötapaukset	11
3.1.2. Lausunnon antajan käyttötapaukset	13
3.1.3. Päätöksen kirjaajan käyttötapaukset	15
3.2. Palvelinohjelmisto	16
3.2.1. PHP	16
3.3. Käytettävyys	17
3.4. Käytettävyyden määritelmien soveltaminen käytännössä	18
3.4.1. Kehykset	19
3.4.2. Valikot	19
3.4.3. Lomakkeet	19
3.4.4. Hakulomakkeet	19
3.4.5. Hakutulostaulukot	20
3.4.6. Virhetilanteet	20
3.5. Käyttöliittymä	20
3.6. Tietoturva	23
3.7. Tunnistautuminen ja käyttöoikeudet	23
4. TOTEUTUS JA TESTAUS	24
4.1. Sovelluksen toteutus	24
4.1.1. Käyttötapaukset	24
4.2. Sovelluksen testaus	30
5. YHTEENVETO	31
6. LÄHDELUETTELO	32
7. LIITELUETTELO	33

KÄYTETYT MERKIT JA LYHENTEET

HTML	Www-sivujen luonnissa käytetty kuvauskieli (Hyper Text Markup Language).
SQL	Tietokantojen käsittelyssä käytetty, IBM:n patentoima ja kehittämä kyselykieli (Structured Query Language)
CSS	Yksinkertainen tyylikieli, jonka avulla määritellään HTML, XHTML ja XML-dokumenttien ulkoasu (Cascading Style Sheets)
JavaScript	Suosituin Internetissä käytettävä skripti-kieli, joka on suunniteltu luomaan HTML-sivuille interaktiivisuutta. Toimii kaikissa web-selaimissa.
Tag	Elementtienmerkkaus, HTML-kieli sisältää elementtejä ja elementtien erottamiseksi asiasisällöstä tarvitaan merkkejä, joita kutsutaan englanninkielellä Tägeiksi.

1. JOHDANTO

Tämä opinnäytetyö tehtiin omana projektinaan, tyydyttämään tilaajaorganisaation todellista tarvetta. Työn valinnassa isoimpana vaikuttavana asiana oli tarve saada opinnäytetyön aiheeksi jokin tuotantoon menevä ohjelmistotuote tai ratkaista todellinen ongelma. Työn lähtökohtana oli hyödyntää tilaajaorganisaation jo olemassa olevia palvelimia ja hankintaprosesseja. Työn tavoitteena oli tuottaa www-pohjainen hankintasovellus, jonka avulla käyttäjät voivat luoda uusia hankintaesityksiä, antaa hankintaesityksiin lausuntoja sekä syöttää hankintaesityksien päätökset sovellukseen ja kyetä seuraamaan esityksen etenemistä hankintaprosessin eri vaiheissa. Tavoiteoptiona oli, ajan riittäessä, toteuttaa myös Active Directory integraatio, jolloin sovelluksen käyttäjä ei tarvitse erillistä tunnistautumista käyttäessään hankintasovellusta. Projektin ulkopuolelle rajattiin sähköinen allekirjoitus.

Opinnäytetyön tekijä toimi projektin resurssina. Projektin taustaorganisaatioon kuului oppilaitoksen puolelta valvova opettaja sekä tilaajaorganisaation puolelta erikseen nimetty joukko tietohallinnon ja hankintatoimiston työntekijöitä.

2. TIETOKANNAN SUUNNITTELU

Suunnitteluvaiheessa hankintasovellus jaettiin kahteen erilliseen osa-alueeseen: tietokantaan ja varsinaiseen sovellukseen. Sovellus jaettiin edelleen kahteen erilliseen osa-alueeseen, palvelinohjelmistoon (PHP-sovellukset) ja käyttöliittymään (HTML, CSS ja JavaScript). Suunnittelu aloitettiin tietokantaosiesta, joka on koko sovelluksen perusta. Tietokantaan tallennetaan kaikki käyttäjien sovellukseen syöttämä data. Tietokannan suunnittelu oli koko työn tärkein osa-alue. Kun suunnittelun epäonnistuminen huomataan, on yleensä sovelluksen muiden osioiden suunnittelu ja toteutus jo pitkällä ja tietokannan muutokset aiheuttavat suuria muutoksia jo valmiiseen ohjelmakoodiin. Tietokannan suunnittelun ja toteutuksen jälkeen voitiin aloittaa muiden osioiden suunnittelu. Käyttöliittymän ja palvelinohjelmiston suunnittelu kyettiin toteuttamaan rinnakkain.

2.1. Yleistä tietokannoista

Tietokantoja käytetään tiedon varastointiin. Nykyisin tietokantoja löytyy lähes kaikkialta, esimerkkinä muun muassa matkapuhelimen puhelinluettelo. Hankintasovellus perustuu MS SQL 2008-tietokantapalvelimelle sijoitettavaan relaatiotietokantaan. Tilaajan tahtotilana oli käyttää MS SQL-tietokantapalvelinsovellusta, vaihtoehtoisena palvelinohjelmistona olisi ollut MySQL-tietokantapalvelinsovellus. MySQL-palvelinsovellus on ns. open licence -sovellus ja näin ollen vapaasti käytettävissä, ilman erillistä korvausta, kun taas MS SQL-palvelinsovellus on Microsoftin tuottama ja lisensoima maksullinen tuote.

Tietokantojen historiaa

Ensimmäiset tietokantakehitykset luotiin 1960-luvulla Charles Backmanin toimesta. Tässä yhteydessä syntyi ensimmäinen toteutettu tietokantamalli, hierarkkinen tietokanta. Hierarkkisessa tietokannassa tiedot järjestettiin hierarkkiseen muotoon. Samassa yhteydessä syntyivät myös ensimmäiset tietokannanhallintajärjestelmät. Hierarkkinen malli muistutti puu-mallia ja sen rakenne oli jäykkä, eikä tiedon toistolta voitu välttyä.

Verkkotietokantamalli korjasi hierarkkisen tietokannan ongelmia vähentäen tiedon toistoa tietokannassa./4, s. 1/

Vuonna 1970 Edgar F. Codd esitteli teoriansa, matemaattiseen logiikkaan perustuvan relaatiomallin, joka poikkesi suuresti vallalla olevista hierarkkisesta tietomallista ja verkkomallista. Seuraavan vuosikymmenen aikana kilpailu tietokantamallien välillä oli kovaa relaatiomallin pysyessä teoriatasolla. Tämän teoriavaiheen pitkän keston ansiosta relaatiotietokannasta tuli käyttäjäystävällinen kun aikaisemmat tietokantamallit oli suunniteltu pitkälti laitteistojen ja ohjelmistojen ehdoilla./4, s. 1/

Vuonna 1980 IBM julkaisi ensimmäisen kaupallisen relaatiomalliin perustuvan tietokannan ja vuonna 1985 julkaistiin ensimmäinen SQL-standardi. Relaatiotietokantojen menestystarina on jatkunut 1980-luvulta aina näihin päiviin saakka./9, s.10/

2.2. Relaatiotietokannat

Relaatiotietokannassa tiedot esitetään relaatioissa, joita yleisemmin kutsutaan tauluiksi. Taulut koostuvat riveistä ja sarakkeista. Rivejä kutsutaan tietueiksi ja sarakkeita kutsutaan kentiksi. Relaatiotietokannassa tieto pyritään tallentamaan siten, että sama tieto esiintyy vain yhdessä taulussa. Relaatiotietokanta sisältää myös tiedon siitä miten taulut liittyvät toisiinsa. Taulujen välille luodaan yhteys eli relaatio, joka nopeuttaa tiedon syöttämistä ja päivittämistä, koska tieto syötetään tai päivitetään vain yhteen tauluun./3, s.1/

2.2.1. Relaationtietokantojen osat

Tietue sisältää tietokantataulun yhden rivin tiedot, kaikki samalla rivillä sijaitsevat tiedot liittyvät samaan asiaan, tapahtumaan tai henkilöön. Tietue jakaantuu useiksi sarakkeiksi eli kentiksi. Saman taulun jokainen tietue sisältää yhtä monta kenttää. Tietokannan yksittäinen tieto tallennetaan kenttään. Kentän tietotyyppi riippuu siihen tallennettavan tiedon tyypistä ja kentän koko riippuu siihen tallennettavan tiedon koosta. Esimerkiksi tallennettaessa asiakkaan etunimi kenttään -kentän tietotyyppi on varchar(10), sulkeissa oleva numero 10

kertoo kuinka monta merkkiä etunimen tallentamiselle varataan. Koska tauluissa ei saa olla kahta täysin identtistä tietuetta, täytyy tietue voida yksilöidä yhden tai useamman kentän avulla. Tällaisia kenttiä kutsutaan avainkentiksi eivätkä ne voi olla tyhjiä. Perus- tai pääavaimeksi valitaan kenttä, joka absoluuttisesti yksilöi tietueen. Usein perus- tai pääavaimena käytetään asiakas- tai tilausnumeroa.

2.3. SQL

Structure Query Language on IBM:n kehittämä standardoitu kyselykieli relaatiotietokannoille, joka ei määrittele verkkoprotokollaa eikä tapaa, jolla kielen käskyjä välitetään. Tärkeimpiä määrittelykäskyjä ovat:

- CREATE DATABASE
- CREATE TABLE
- CREATE VIEW.

Tiedon täyttökäskyistä tärkeimpiä ovat:

- SELECT
- UPDATE
- INSERT
- DELETE.

SQL-kielellä tehdyissä komentojonoissa itse SQL-kielen komennoissa ei isoilla ja pienillä kirjaimilla ole merkitystä, kun taas tietokannan, taulun ja kentän nimissä isot ja pienet kirjaimet omaavat eri merkityksen.

Tietokannan luominen

Tietokanta luodaan tietokannan hallintaohjelmistolla suoritettavalla SQL-kielisellä käskyllä. Ensimmäiseksi luodaan tietokannan taulut joihin ei ole viittauksia, toisin sanoen ne sijaitsevat ER-kaavion öreunoilla. Tietokannan ja taulujen luontiin käytettävä SQL-kielinenä käsky on CREATE, jota käytetään sekä tietokannan että taulun luomisessa.

Tietokannan muokkaaminen

Tietokannan muokkaamiseen käytetään myös SQL-kielisiä käskyjä, jotka suoritetaan hallintaohjelmiston konsolilta tai asiakassovelluksessa. Tietokannan muokkaamiseen

yleisimmin käytetyt SQL-kieliset käskyt ovat UPDATE ja INSERT. INSERT-käsky lisää tauluun yhden tietueen (rivin). UPDATE-käsky päivittää käskyssä määritellyn kentän sisällön ja jos UPDATE-käskyn yhteydessä ei käytetä WHERE-käskyä, kohdistuu se kaikkiin tietueisiin. WHERE-komennolla voidaan asettaa ehto joka määrää mihin tietueeseen/tietueisiin käsky kohdistuu.

Tietokantakyselyt

Tietokannasta haetaan tietoa tietokantakyselyillä jotka muodostuvat SQL-kielisistä käskyistä. Kyselyt suoritetaan joko hallintaohjelmiston konsolilta (Liite 1) tai asiakassovelluksessa. Tietokanta hauissa yleisimmin käytetty SQL-käsky on SELECT. Seuraavassa on tietokantahaku avattu esimerkin avulla:

```
SELECT nimi FROM tkanta.asiakkaat WHERE asiakasno = 200
```

Haussa haetaan tietokannan *tkanta*-taulusta *asiakkaat* sen tietueen sisältö kentästä *nimi*, minkä *asiakasno* kentässä on tieto 200.

MS SQL 2008 Server

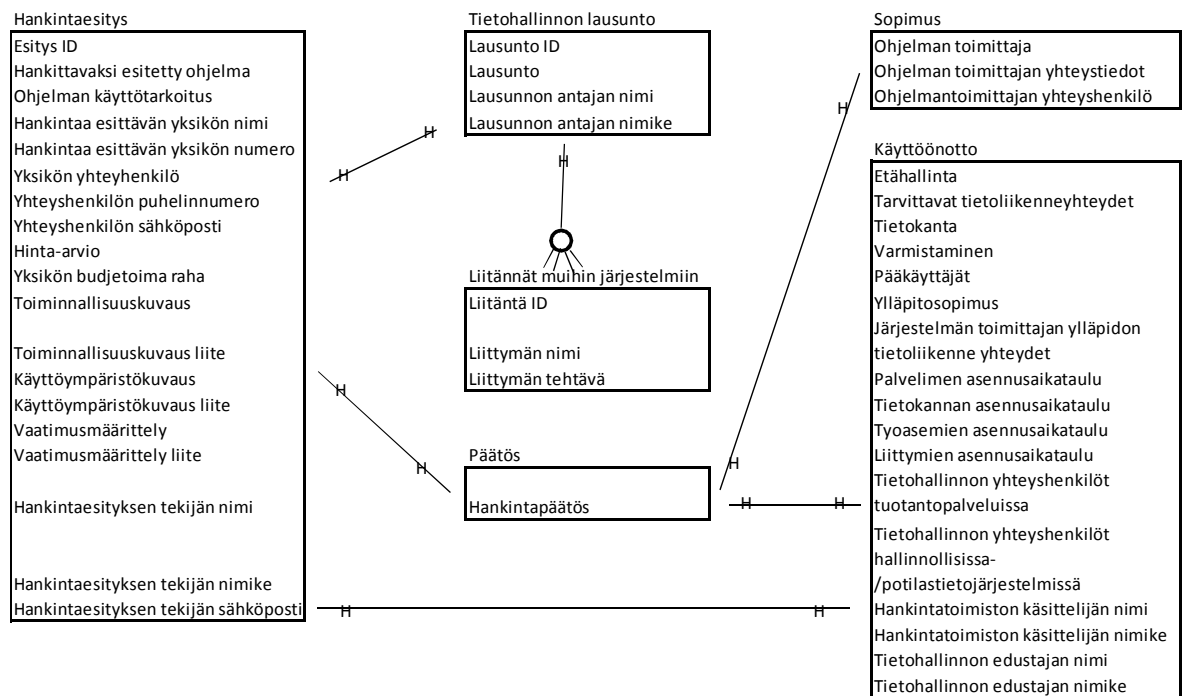
Tässä projektissa käytettiin Microsoftin SQL 2008 -tietokantapalvelinohjelmistoa. MS-SQL:n käyttämä SQL-kielen syntaksi, eroaa hiukan MySQL-ohjelmiston vastaavasta. Käskyt ovat samat, ainoastaan kieliopit poikkeavat toisistaan. MS SQL 2008-ohjelmisto tarjoaa tietokantojen hallintaan SQL Server Management Studio-hallintaohjelmiston, myös tässä projektissa sitä käytettiin tietokannan ja taulujen luontiin.

2.4. Hankintasovelluksen tietokannan suunnittelu

Hankintasovelluksen tietokannan määrittely aloitettiin olemassa olevien lomakkeiden pohjalta. Lomakkeista saatiin tieto käyttäjien sovellukseen syöttämästä tiedosta, lisäksi tietokannan suunnittelussa käytettiin prosessikuvausta nykyisestä prosessista. Kerättävän tiedon ja prosessikuvauksen perusteella aloitettiin tietokantasuunnittelu./2, s.15619/

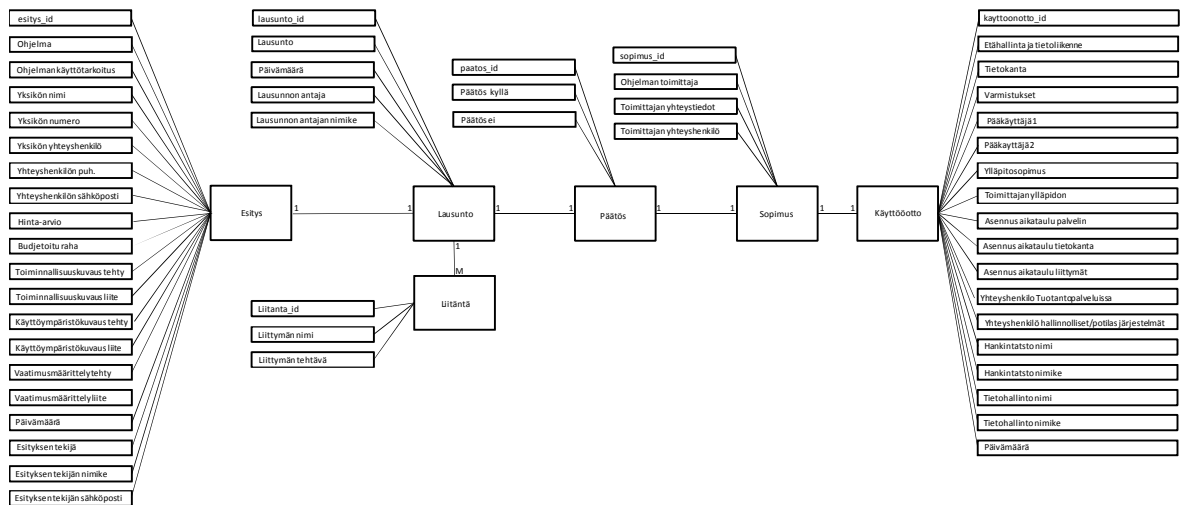
Tietokantasuunnittelu aloitettiin tekemällä käsiteanalyysi, jossa suunnitellaan tietokantaa loogisella tasolla ja jonka perusteella luodaan käsitemalli, jonka luonnin perusideoina ovat seuraavat asiat:

- käsitteiden tunnistaminen
- yhteyksien muodostaminen
- tietojen lisääminen
- monesta ó moneen yhteyksien purkaminen
- käsitemallin kuvaaminen.



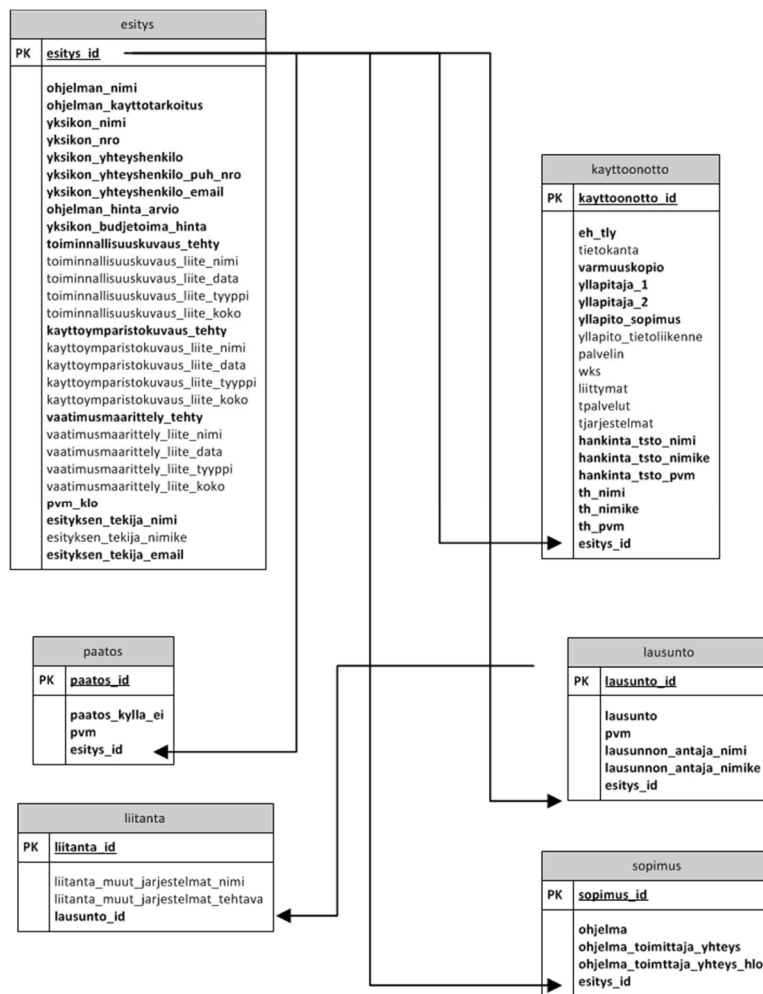
Kuva 1. Hankintasovelluksen tietokannan käsitemalli

Käsitemalli on karkea versio tietokannasta ja sen perusteella luotiin ER-kaavio, joka sisältää tietokannan taulurakenteen. ER-malli kuvaa tietokannassa kuvattavaa kohdealuetta, tarkoituksena määrittää tietokannan fyysisen rakenteen perusta (Kuva 1). Muutettaessa ER-kaaviota relaatiotietokantamalliksi, käsitteistä tulee tauluja ja tiedoista taulujen yksittäisiä tietueita eli sarakkeita. Myös taulujen väliset riippuvuudet määräytyivät käsiteanalyysin perusteella. Tuloksena sovelluksen käyttämä relaatiotietokanta (Kuva 2).



Kuva 2. Hankintasovelluksen tietokantasuunnittelun ER-kaavio

ER-kaavion perusteella luotiin tietokannan taulurakenne ja taulujen väliset yhteydet (Kuva 3). Johtuen MS SQL-tietokannan tavasta käsitellä binääritiedostoja, jouduttiin käsittemallista poiketen luomaan esitystauluun muutamia lisätietueita. Binääritiedosto jouduttiin muuntamaan, PHP-kielen funktiota hyväksi käyttäen, heksadesimaalimuotoon jotta se kyettiin tallentamaan tietokantaan. Tämän lisäksi tietokantaan tallennettiin tiedot tiedoston nimestä, tiedostotyyppistä ja tiedoston koosta.

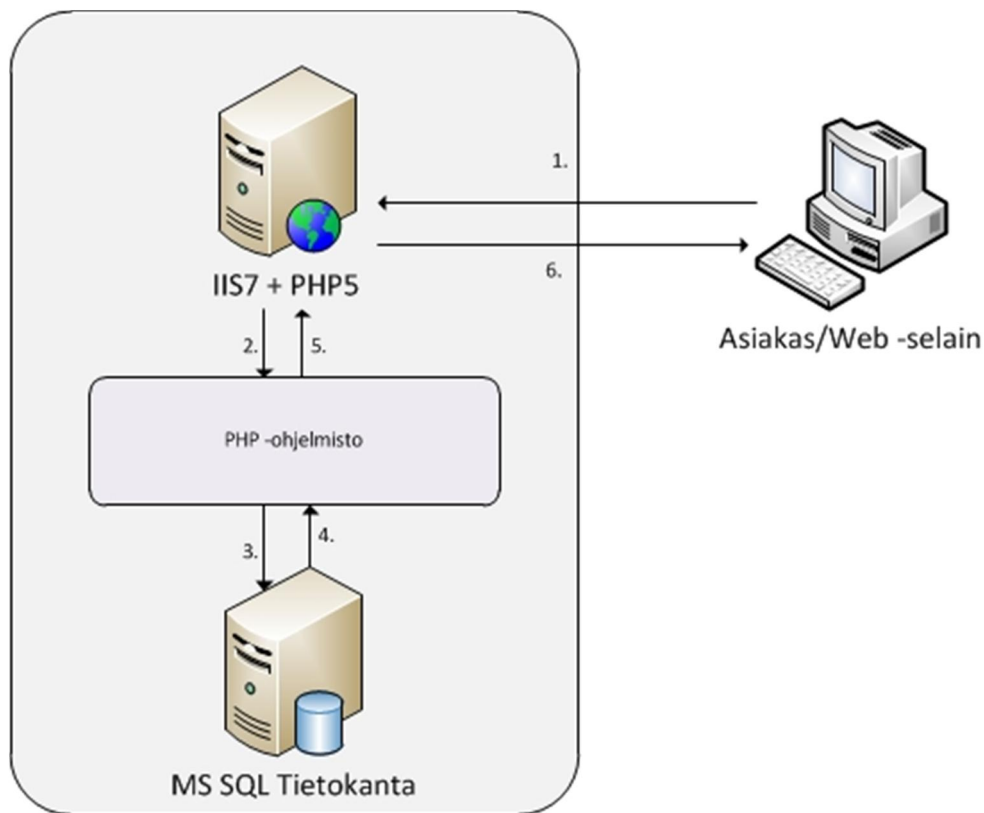


Kuva 3. Hankintasovelluksen tietokannan taulut

Tietokanta luotiin käyttäen MS SQL -palvelimen mukana tulevaa MS SQL Server Management Studio -ohjelmaa. Tietokanta luotiin käyttäen ohjelman graafista työkalua. Käyttöoikeudet tietokantaan annettiin luomalla paikallinen käyttäjä MS SQL -palvelimeen, jolle annettiin MS SQL -Server tasolla dbcreator, public ja sysadmin oikeudet MS SQL -palvelimeen. Tietokantataulut luotiin käyttämällä SQL-kielistä scriptejä (Liite 1).

3. WWW-SOVELLUKSEN SUUNNITTELU

Hankintasovelluksen palvelimella suoritettavat osat toteutettiin PHP-ohjelmointikielellä. PHP-kielinen koodi suoritetaan www-sivua ladattaessa palvelimella. Www-palvelimella on oltava PHP-ohjelmisto asennettuna. Tässä projektissa käytettiin www-palvelimena Microsoftin IIS 7-palvelinta johon asennettiin PHP ohjelmiston 5.3 versio ja Microsoftin omat MSSQL laajennukset MS SQL -yhteyksiä varten. Käyttöliittymän dynaamisuutta ja interaktiivisuutta lisäämään käytettiin www-sivuilla myös JavaScriptiä. JavaScript eroaa PHP:stä suorituspaikan perusteella, PHP:tä suoritetaan palvelimella ja JavaScripti suoritetaan selaimessa. JavaScript ei lähetä tietoja palvelimelle.



Kuva 4. www-sovelluksen arkkitehtuuri kuvaus

Kuvan 4 toimintojen selitykset:

1. Web-selain lähettää www-palvelimelle sivupyynnön.
2. Www-palvelin tunnistaa sivun PHP-tarkenteen ja lähettää sivun PHP-ohjelmistolle tulkittavaksi.

3. PHP-ohjelmisto suorittaa sivulla mahdollisesti olevat tietokantahaut.
4. Tietokantapalvelin vastaa mahdollisiin hakuihin.
5. PHP-ohjelma tuottaa ohjelmakoodin ja mahdollisten tietokantahakujen tuottamien tietojen perusteella www-palvelimelle HTML-sivun.
6. Www-palvelin palauttaa HTML-sivun web-selaimelle.

3.1. Käyttötapaukset

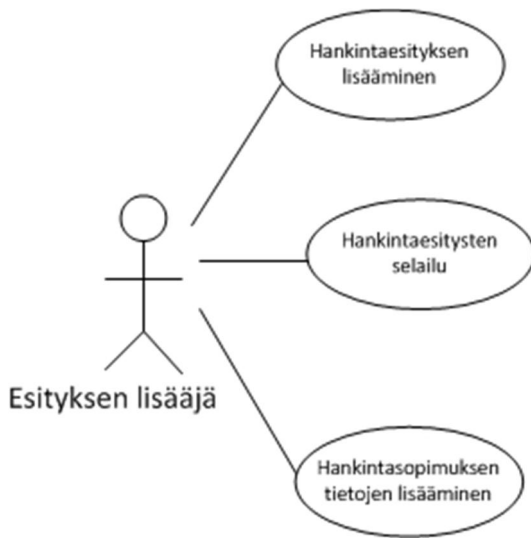
Käyttötapaukset kuvaavat käyttäjän ja sovelluksen välistä vuorovaikutusta ja ne kuvataan sanallisesti tai vaihtoehtoisesti kaavioina. Käyttötapauskuvaukset voivat sisältää seuraavat tiedot:

- käyttäjä
- käyttövaatimukset
- tapahtumien kulku
- esiehdot
- poikkeukset
- virhetilanteet
- lopputulos.

Hankintasovelluksen suunnittelussa käyttötapaukset suunniteltiin kolmen eri käyttäjän kannalta:

- esityksen tekijä
- lausunnon antaja
- päätöksen kirjaaja.

3.1.1. Esityksen tekijän käyttötapaukset



Kuva 5. Käyttötapaus esityksen tekijä

Esityksen tekijän käyttötapauksen kuvaukset (Kuva 5).

Hankintaesityksen lisäys

Käyttötapaus: Hankintaesityksen lisääminen

Käyttövaatimus: Käyttäjän tulee kyetä lisäämään uusi hankintaesitys, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet lisätä uusi hankintaesitys järjestelmään.

Tapahtumien kulku: Käyttäjä klikkaa hiirellä öUusi hankintaesitysö-linkkiä ja syöttää vaaditut tiedot avautuvaan lomakkeeseen.

Poikkeukset/virhetilanteet: Yhteys tietokanta palvelimeen tai LDAP-hakemiston ja www-palvelimen välillä ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia hankintaesityksen lisäämiseen.

Lopputulokset: Käyttäjä lisää uuden hankintaesityksen.

Hankintaesitysten hakeminen

Käyttötapaus: Hankintaesitysten haku

Käyttövaatimus: Käyttäjän tulee kyetä selaamaan tehtyjä hankintaesityksiä, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet selata tehtyjä hankintaesityksiä.

Taphtumien kulku: Käyttäjä klikkaa hiirellä öHae hankintaesitysö-linkkiä ja valitsee avautuvasta valikosta haun perusteen (esityksen numeron, tuotteen nimen, esityksen tekijän nimen tai kaikki esitykset) ja antaa tarvittavat tiedot hakulomakkeelle.

Poikkeukset/virhetilanteet: Yhteydet tietokantapalvelimeen ja LDAP-palvelimeen www-palvelimelta ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia hankintaesityksien selailuun.

Lopputulos: Käyttäjä selaa tehtyjä hankintaesityksiä.

Tietojen lisääminen hankintasopimukseen

Käyttötapaus: Hankintasopimuksen tietojen lisääminen

Käyttövaatimus: Käyttäjän tulee kyetä lisäämään hankintasopimuksen tiedot hankintaesitykseen, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet lisätä hankintasopimuksen tiedot sovellukseen.

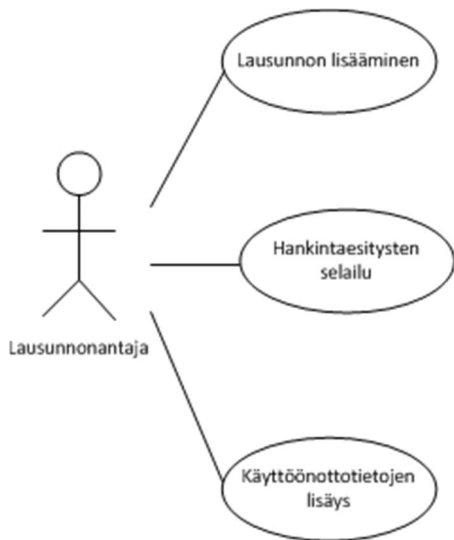
Taphtumien kulku: Käyttäjä klikkaa hiirellä öAnna sopimuksen tiedotö-linkkiä ja syöttää vaaditut tiedot avautuvaan lomakkeeseen.

Poikkeukset/virhetilanteet: Yhteys tietokantapalvelimeen tai LDAP-hakemiston ja www-palvelimen välillä ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia hankintaesityksen lisäämiseen.

Lopputulos: Käyttäjä lisää hankintasopimuksen tiedot sovellukseen.

Käyttötapausluettelossa ei ole esitysten, tietojen tai päätösten poistomahdollisuutta, koska käyttäjillä ei ole mahdollisuutta poistaa tietokannasta tietoja. Tietojen poistaminen on mahdollista vain tietokannan ylläpidon toimesta, tilaajatahon toimintamallien mukaisesti.

3.1.2. Lausunnon antajan käyttötapaukset



Kuva 6. Käyttötapaus lausunnon antaja

Lausunnon antajan käyttötapaukset (Kuva 6).

Lausunnon lisäys esitykseen

Käyttötapaus: Lausunnon lisäys.

Käyttövaatimus: Käyttäjän tulee kyetä lisäämään hankintaesitykseen lausunto, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet lisätä lausunto hankintaesitykseen.

Tapahtumien kulku: Käyttäjä klikkaa hiirellä öAnna lausunto esitykseenö-linkkiä ja hakee esityksen tiedot esitysnumeron perusteella ja syöttää lausunnon lomakkeeseen.

Poikkeukset/virhetilanteet: Yhteydet tietokantapalvelimeen ja LDAP-palvelimeen www-palvelimelta ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia lausunnon lisäämiseen.

Lopputulokset: Käyttäjä lisää lausunnon hankintaesitykseen.

Esitysten haku ja selaaminen

Käyttötapaus: Hankintaesitysten selailu

Käyttövaatimus: Käyttäjän tulee kyetä selaamaan tehtyjä hankintaesityksiä, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet selata tehtyjä hankintaesityksiä.

Tapahtumien kulku: Käyttäjä klikkaa hiirellä öHae hankintaesitysö-linkkiä ja valitsee avautuvasta valikosta haun perusteen (esityksen numeron, tuotteen nimen, esityksen tekijän nimen tai kaikki esitykset) ja antaa tarvittavat tiedot hakulomakkeelle.

Poikkeukset/virhetilanteet: Yhteydet tietokantapalvelimeen ja LDAP-palvelimeen www-palvelimelta ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia hankintaesityksien selailuun.

Lopputulos: Käyttäjä selaa tehtyjä hankintaesityksiä.

Käyttöönottotietojen lisääminen esitykseen

Käyttötapaus: Käyttöönottotietojen lisäys.

Käyttövaatimus: Käyttäjän tulee kyetä selaamaan tehtyjä hankintaesityksiä, käyttäen omaa toimialueeseen liitettyä työasemaa.

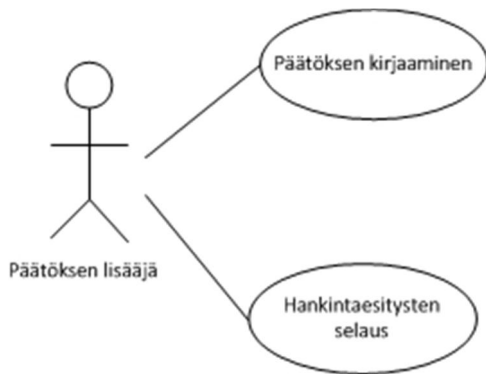
Esiehdot: Käyttäjällä tulee olla oikeudet selata tehtyjä hankintaesityksiä.

Tapahtumien kulku: Käyttäjä klikkaa hiirellä öAnna käyttöönottotiedotö-linkkiä ja hakee esityksen esitysnumeron perusteella ja lisää käyttöönottotiedot lomakkeeseen.

Poikkeukset/virhetilanteet: Yhteydet tietokantapalvelimeen ja LDAP-palvelimeen www-palvelimelta ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia hankintaesityksien selailuun.

Lopputulos: Käyttäjä lisää käyttöönnotossa tarvittavat tiedot hankintaesitykseen.

3.1.3. Päätöksen kirjaajan käyttötapaukset



Kuva 7. Käyttötapaus päätöksen lisääjä

Kirjaajan käyttötapaukset (Kuva 7).

Päätöksen kirjaaminen esitykseen

Käyttötapaus: Päätöksen kirjaaminen.

Käyttövaatimus: Käyttäjän tulee kyetä lisäämään päätös, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet lisätä päätös sovellukseen.

Tapahtumien kulku: Käyttäjä klikkaa hiirellä öSyötä päätösö-linkkiä ja täyttää avautuvaan lomakkeeseen tarvittavat tiedot.

Poikkeukset/virhetilanteet: Yhteydet tietokantapalvelimeen ja LDAP-palvelimeen www-palvelimelta ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia päätöksen lisäämiseen.

Lopputulokset: Käyttäjä saa lisätä päätöksen sovellukseen.

Hankintaesitysten haku ja selaaminen

Käyttötapaus: Hankintaesitysten selailu

Käyttövaatimus: Käyttäjän tulee kyetä selaamaan tehtyjä hankintaesityksiä, käyttäen omaa toimialueeseen liitettyä työasemaa.

Esiehdot: Käyttäjällä tulee olla oikeudet selata tehtyjä hankintaesityksiä.

Tapahtumien kulku: Käyttäjä klikkaa hiirellä öHae hankinta esityso-linkkiä ja valitsee avautuvasta valikosta haun perusteen (esityksen numeron, tuotteen nimen, esityksen tekijän nimen tai kaikki esitykset) ja antaa tarvittavat tiedot hakulomakkeelle.

Poikkeukset/virhetilanteet: Yhteydet tietokantapalvelimeen ja LDAP-palvelimeen www-palvelimelta ovat poikki tai käyttäjällä ei ole riittäviä oikeuksia hankintaesityksien selailuun.

Lopputulokset: Käyttäjä selaa tehtyjä hankintaesityksiä.

3.2. Palvelinohjelmisto

Palvelinohjelmiston koodauksessa käytettiin PHP-kieltä. PHP-ohjelmointikieli on PERL-ohjelmointikielen kaltainen komentosarjakieli, joka tulkitaan vasta suorituksen aikana ja jota käytetään www-palvelinympäristössä dynaamisten www-sivujen luontiin. Kielen kehittämisen aloitti Rasmus Lerdorf vuonna 1994. Vuonna 1997 julkaistiin PHP 2.0. Vuonna 1998 kirjoitettiin PHP 2.0 uudelleen ja sen seurauksena syntynyt PHP 3.0 on hyvin pitkälti samankaltainen kuin nykyiset versiot PHP:sta. Toukokuussa 2000 julkaistiin PHP 4.0 joka perustui uuteen ytimeen, nimeltään Zend Engine. Viimeisin versio PHP 5.0 julkaistiin heinäkuussa 2004 ja uusin vakaa versio on 5.3.10. Nykyisellään PHP-kieli on maailman eniten käytetty dynaamisten www-sivujen tuottamiseen käytetty ohjelmointikieli./8, s.1/

3.2.1. PHP

PHP tarjoaa useita erilaisia tapoja upottaa ohjelmakoodia www-sivun lähdekoodiin. Yleisin tapa on upottaa PHP-koodi www-sivun lähdekoodiin, on käyttää öpapukaijamerkkejäö.

Esimerkki koodi:

1. <body>
2. <?php
3. echo (öYleisin tapa upottaa php-koodia dokumenttiinö);
4. ?>

5. </body>

Php-ohjelmakoodissa käytetään yleisesti include-komentoa, jonka avulla voidaan sivun ohjelmakoodiin sisällyttää toinen ohjelmakoodia sisältävä tiedosto. Esimerkiksi:

1. <?php
2. include(ödb\connect_db.phpö);
3. ?>

Näin voidaan käyttää connect_db.php tiedoston sisältämiä php-kielisiä funktioita käyttäen php-tagien välissä, ihan kuin ne sijaitisivat samassa tiedostossa. Tämä on hyväksi havaittu tapa, kun samoja funktioita käytetään useissa eri tiedostoissa, vältetään saman koodin toistolta.

Hankintasovelluksen PHP-ohjelmointikielellä toteutetut ohjelmat jaettiin käyttötapauksen perusteella pienempiin moduuleihin. Moduulit sijaitsevat www-palvelimella omilla hakemistoissaan. Käyttötapauksen toteuttamiseen tarvittavien ohjelmanosien lisäksi tarvittiin kaikkien käyttötapauksen yhteydessä käytettäviä ohjelman osia, esimerkiksi tietokantayhteyden luomiseen ja purkamiseen hallitusti. Tietokantayhteyden muodostamiseen tarvittava käyttäjätunnus ja salasana säilytetään erillisissä tiedostoissa, jotka eivät sijaitse www-palvelimen www-sivustojen kanssa samassa hakemistohaarassa.

Palvelinohjelmisto suunniteltiin modulaariseksi, jolloin jokainen toiminnallisuus on omana moduulina. Rakenne mahdollistaa muutosten joustavan hallinnan ja helpon tavan sovelluksen päivittämiseen ja mahdollisten myöhemmin suunniteltavien lisäominaisuuksien käyttöönottoon (Liite 2).

3.3. Käytettävyys

Käytettävyydelle löytyy useita eri määritelmiä ja eri tahot käsittävät sen eri tavalla. Web-maailmassa yksi tunnetuimmista asiantuntijoista on Jacob Nielsen. Nielsen määrittelee

käytettävyyden laadun määritelmänä, joka mittaa, kuinka helppoa sovelluksen käyttö on. Nielsen on määritellyt myös käytettävyydelle komponentit:

- Opittavuus: Kuinka helposti käyttäjä osaa perusasiat ensimmäisellä käyttökerralla.
- Tehokkuus: Kun asiat osataan, kuinka nopeasti käyttäjä kykenee toteuttamaan tehtävät.
- Muistettavuus: Kuinka kauan käyttäjiltä menee saavuttaa sama tuottavuus käyttötaun jälkeen.
- Virheettömyys: Kuinka paljon käyttäjät tekevät virheitä ja kauanko niistä palautuminen kestää.
- Miellyttävyys: Kuinka miellyttävä tuote/sovellus on käyttää.
- Hyödyllisyys: Kuinka hyvin tuote/sovellus sopii työhön johon se on suunniteltu.

/7, s.1/

Lisäksi ISO standardi sisältää hyvän käytettävyyden määrittelyn ISO 9241611. Sen määrittelemät käytettävyyden komponentit ovat:

- Vaikuttavuus: Kuinka hyvin ja ajallaan tehtävät saadaan suoritettua.
- Tehokkuus: Paljonko resursseja kuluu tavoitteen saavuttamiseksi.
- Miellyttävyys: Kuinka miellyttäväksi käyttäjä tehtävän suorittamisen kokee.

/5, s.8/

3.4. Käytettävyyden määritelmien soveltaminen käytännössä

Hankintasovelluksen käyttöliittymässä on pyritty ottamaan huomioon hyvän käytettävyyden määrittelyitä. Sivujen värimaailma seuraa PPSHP:n www-sivujen väritystä, lähtien taustan väristä ja päätyen tekstin väriin. Toimintoja pyrittiin automatisoimaan käyttäen JavaScriptiä, jolloin välttyttiin suorittamasta tarkastuksia ja automatisointeja PHP-ohjelmistolla ja näin saatiin palvelimen kuormitusta alenemaan.

3.4.1. Kehykset

Kehysten avulla sovelluksen yleisnäkymä kyettiin pitämään selkeän yksinkertaisena. Kahden kehyksen malliin päädyttiin sen yksinkertaisuuden ja selkeyden vuoksi. Vasen kehys toimii valikkojen sijoituspaikkana ja oikea kehys sisältää lomakkeet ja hakutulokset. Kehysten väliset reunukset piilotettiin, jolloin kehysten näkyvyys käyttäjälle vähenee. Kehysten koon muuttaminen estettiin, jolloin yleisnäkymä säilyy joka tilanteessa samana.

3.4.2. Valikot

Valikot on toteutettu taulukoiden avulla, jolloin niiden asemointi on saatu säännöllisen näköiseksi. Valikon linkeistä on poistettu käytetyn linkin värit, koska kyseessä on sovellus, jolloin jokaista linkkiä voidaan käyttää useasti, eikä tieto jo vierailusta sivusta anna lisäarvoa käyttäjälle. Värit vastaa taustaltaan muualla sivustossa tekstin väritystä ja aktiivinen linkki on valkoinen, jotta hiiren sijainti erottuu selkeästi tummaa taustaa vasten.

3.4.3. Lomakkeet

Lomakkeiden ulkoasut on tuotettu tyylitiedoston ja HTML-kielen avulla. Lomakkeessa olevien asiakenttien ryhmittely on aikaan saatu HTML-kielen `<fieldset>` `</fieldset>`-tageilla ja ryhmille otsikot `<legend>``</legend>`-tageilla. Lomakkeiden ryhmittelyllä on pyritty parantamaan lomakkeen hahmottamista, mitä paremmin käyttäjä hahmottaa lomakkeen, sitä suuremmalla todennäköisyydellä käyttäjä myös täyttää lomakkeen oikein./1, s.260/

JavaScriptin avulla tarkistetaan pakolliset kentät ennen tietojen lähettämistä, näin helpotetaan käyttäjän toimia, mikäli käyttäjä on unohtanut täyttää pakollisen kentän JavaScript huomauttaa siitä jolloin käyttäjän ei tarvitse kirjoittaa kaikkia lomakkeen tietoja uudelleen. Muuten kentän sisältämään dataan ei oteta kantaa.

3.4.4. Hakulomakkeet

Haettaessa tietoa tietokannasta voidaan hakukriteeri valita esityksnumeron, esityksen tekijän tai tuotteen perusteella, lisäksi voidaan tulostaa kaikki hankintaesitykset. Jokaista hakua

varten on oma hakulomake joka avautuu valikosta valitsemalla. Hakulomakkeet pyrittiin pitämään mahdollisimman yksinkertaisena, jolloin niiden käyttö on mahdollisimman helppoa. Lomakkeiden värit noudattelee samaa värimaailmaa kuin muukin sivusto.

3.4.5. Hakutulostaulukot

Hakujen tulokset näytetään taulukkomuodossa, haun tuloksena saadaan vain esityksen perustiedot, mutta avaamalla tuloksissa olevia linkkejä, päästään tarkastelemaan esityksen tarkempia tietoja ja mahdollisia liitetiedostoja. Taulukon värit noudattelevat muun sivuston värimaailmaa.

3.4.6. Virhetilanteet

Sovelluksen käytön virhetilanteet on pyritty minimoimaan selkeällä ja yksinkertaisella käyttöliittymällä. Käyttäjän antamien väärin syöttötietojen vaikutusten minimoimiseksi on lomakkeisiin toteutettu sisällön oikeellisuuden tarkastuksia JavaScriptillä. Mahdollisten virhetilanteiden käsittely ja niistä toipuminen on toteutettu käyttäjien ohjeistamisella ja sovelluksen tuottamilla virheilmoituksilla.

3.5. Käyttöliittymä

Hankintasovelluksen käyttöliittymä perustuu HTML-kielellä toteutettuihin www- sivuihin, joille on luotu dynaamisuutta selaimessa suoritettavilla pienillä JavaScript-koodeilla. Sivujen sisältöä tuotetaan myös palvelimella suoritettavilla PHP-kielisillä ohjelmilla, jotka hakevat sisältöä sivuille MS SQL -tietokannasta.

JavaScript

JavaScriptin ensimmäinen versio 1.0 julkaistiin vuonna 1995 Netscape Navigator 2 selaimen mukana. Netscape kehitti JavaScriptin korvaamaan kaikki erilaiset CGI-kielet. JavaScriptin alkuperäinen nimi oli LiveScript, kun Sunsoft, Java-ohjelmointikielen kehittäjä ja Netscape pääsivät sopimukseen Java-nimen käytöstä, korvattiin LiveScript nimi JavaScript nimellä. JavaScript on oliopohjainen scriptikieli, joten se sisältää ennalta

määriteltyjä olioita joita voidaan käyttää luotaessa www-sivuja JavaScriptillä. Scripti tarkoittaa tässä yhteydessä, että JavaScript on tulkittava kieli jolloin www-selain osaa tulkita ja käsitellä ohjelmakoodin suoraan ilman kääntämistä./6, s. 1-7/

JavaScriptin avulla sivuille luodaan toiminnallisuuksia joiden toteuttaminen ei onnistu pelkän HTML-kielen avulla, esimerkiksi lomakkeen osien piilottaminen käyttäjän valintojen perusteella tai kehysten sisällön käsittely käyttäjän valitessa valikosta linkin.

CSS tyylitiedostot

HTML-kielen avulla saadaan luotua www-sivujen rakenne ja CSS:llä määritellään sivujen ulkoasu. CSS-tiedosto on tekstitiedosto, joka sisältää sääntöjä ja arvoja joiden perusteella määritellään www-sivujen elementtien ulkoasut. Jokaisella tyylitiedoston tyylisäännöllä on kaksi perusosaa: valitsija ja määritys. Valitsija määrittelee säännön vaikutuspiiriin kuuluvat elementit ja määritys osa sisältää yhden tai useamman ominaisuus/arvo parista, joiden mukaan vaikutuspiiriin kuuluvien elementtien ominaisuudet muuttuvat. Esimerkki tyylisäännöstä:

```
h3 {color:red;background:blue;}
```

Esimerkissä h3 on valitsin joka kertoo säännön vaikutuspiiriin. Vaikutuspiiriin kuuluvat kaikki kolmannen tason otsikot. Hakasulkeiden sisällä ensimmäinen pari color:red, kertoo että kolmannen tason otsikoiden väri on punainen. Toinen pari kertoo että kolmannen tason otsikon taustavärinä käytetään sinistä. ominaisuus/arvo parit erotetaan toisistaan hakasulkeiden sisällä puolipisteellä. CSS-tyylimäärittelyt voidaan ottaa käyttöön joko sisäisten- tai ulkoisten tyylisivujen avulla. Käytettäessä sisäisiä tyylisivuja, sijoitetaan tyylimäärittelyt HTML-tiedoston yläosaan. Luotaessa sisäistä tyylisivua, sijoitetaan HTML-dokumentin head-osioon teksti <style type =text/css> . Tämän jälkeen luodaan tyylisäännöt ja suljetaan tyylissivu kirjoittamalla </style>. /1, s. 120-125 ja 131/

Mikäli samaa tyylitiedostoa aiotaan käyttää myös muissa sivustoissa, on kannattavampaa käyttää ulkoista tyylisivua. Käytettäessä ulkoista tyylitiedostoa tallennetaan tyylisäännöt teksti tiedostoon www-palvelimelle, nimetään tiedosto ja annetaan sille tarkentimeksi .css.

Käyttöön kyseinen tyylitiedosto otetaan sijoittamalla alla oleva teksti HTML-dokumentin head-osioon:

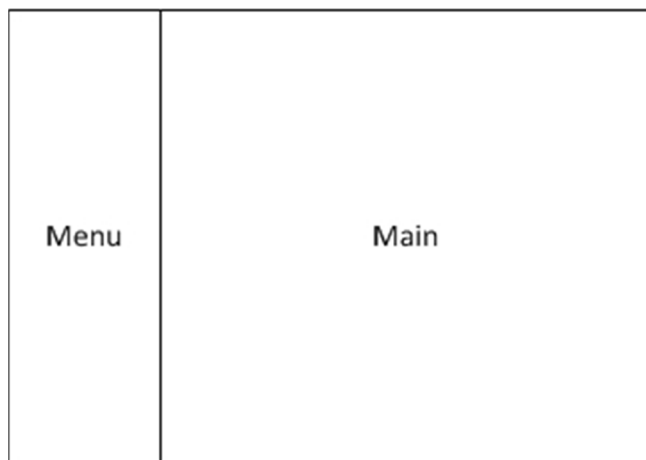
`<link rel=östylesheet type=ötext/css href=url.css/>` url.css kertoo käytettävän tyylitiedoston nimen./1, s. 130/

Käytettäessä ulkoista tyylitiedostoa voidaan tyylitiedosto ottaa käyttöön useissa eri HTML-dokumenteissa lisäämällä yllä oleva teksti dokumenttiin. Tämä mahdollistaa helpon tyylin muuttamisen koko www-sivustolla (Liite 3).

Hb.

HTML

Käyttöliittymä perustuu HTML-kielellä luotuihin www-sivuihin. Etusivulla käytettiin kehystekniikkaa, jolloin etusivu jakaa näytön kahteen kehykseen. Kumpikin kehys voi sisältää erillisen www-sivun.



Kuva 8. Etusivun kehysjako

Vasemmassa reunassa oleva kehys on nimetty Menuksi, tämä kehyksen sisältönä on valikko, josta valitaan halutut toiminnot. Oikean puolen kehys on nimeltään Main. Tähän kehykseen avautuvat kaikki syöttö-, hakulomakkeet ja tulostaulukot (Kuva 8).

3.6. Tietoturva

Sovellus tulee käyttöön sisäverkkoon ja siihen ei ole yhteyksiä Internetistä joten palomuuori osaltaan hoitaa isoimman osan tietoturvasta. Koska sovellus käyttää Windowsin käyttäjätunnusta ja salasanaa suoraan Active Directorysta, myöskään SQL-injektio ei ole iso uhka. Riskien minimoimiseksi lomakkeiden määrättyjen osien syöttötiedot suodatetaan ja niistä poistetaan erikoismerkit sekä mahdollisuus syöttää SQL-komentoja.

3.7. Tunnistautuminen ja käyttöoikeudet

Hankintasovelluksen käyttöoikeudet määräytyvät Windows toimialueen ryhmäjäsenyyksien mukaan. Sovelluksen käyttöoikeudet on jaettu kolmeen osaan:

- Käyttäjä (esityksen tekijä ja sopimustietojen kirjaaja)
- Lausunnon antaja (käyttöönottotietojen kirjaaja)
- Päätöksen kirjaaja.

Kun IIS-palvelimessa otetaan Windows-käyttäjätunnistus käyttöön, käyttäjän työasema on toimialueen jäsen ja hankintasovelluksen FQDN-nimi on määritelty Internet selaimessa luotetuksi alueeksi, saadaan tieto käyttäjän toimialueelle kirjautumiseen käyttämästä käyttäjätunnuksesta välitettyä PHP-ohjelmistolle. PHP-ohjelma tekee LDAP-kyselyn Active Directoryyn ja tarkistaa tunnuksen käyttöoikeudet ryhmäjäsenyyden perusteella. Mikäli käyttäjällä ei ole valitsemansa toiminnon vaatimia oikeuksia, saa käyttäjä virheilmoituksen.

4. TOTEUTUS JA TESTAUS

Työhön kuului sovelluksen testiversion toteutus ja sen alustava testaus kehitysympäristössä. Työn tuloksena syntyi tuotantoympäristössä testattava versio hankintasovelluksesta.

4.1. Sovelluksen toteutus

Sovelluksen toteutus aloitettiin samaan aikaan kuin suunnittelukin. Toteutusvaiheen ensimmäinen tehtävä oli kehitysympäristön asentaminen. Kehitysympäristö asennettiin virtuaalisille koneille, joiden alustana toimi työasemassa toimiva MS Virtual PC -ohjelmisto. MS Virtual PC -ohjelmiston avulla voidaan luoda niin monta virtuaalista työasemaa tai palvelinta kuin fyysisessä työasemassa riittää muistia ja levytilaa. Tässä projektissa tarvitsimme kaksi virtuaalista palvelinta. Toiseen palvelimeen asennettiin MS SQL 2008 -tietokantapalvelin ja toinen palvelin toimi sekä toimialueen ohjauskoneena että IIS 7.0 www -palvelimena. Tällä kehitysympäristön kokoonpanolla kyettiin simuloimaan tilaajan tuotantoympäristöä ohjelmistokehityksen vaatimalla tasolla.

4.1.1. Käyttötapaukset

Käyttötapaukset kertovat millaisia toiminnallisuuksia sovelluksessa tulee olla. Hankintasovelluksessa joukko HTML-, JavaScript- ja PHP-tiedostoja muodostavat eri käyttötapausten edellyttämät toiminnallisuudet.

Hankintaesityksen lisääminen

Uuden hankintaesityksen luomiseen tarvittavat ohjelmat ja tiedostot sijaitsevat www-palvelimella omassa alihakemistossaan. Päävalikosta löytyy valinta öUusi hankintaesitysö, joka valitsemalla avautuu main-kehikseen lomake, jonka täyttämällä ja lähettämällä lomakkeen tiedot tallentuvat tietokantaan (Kuva 9).

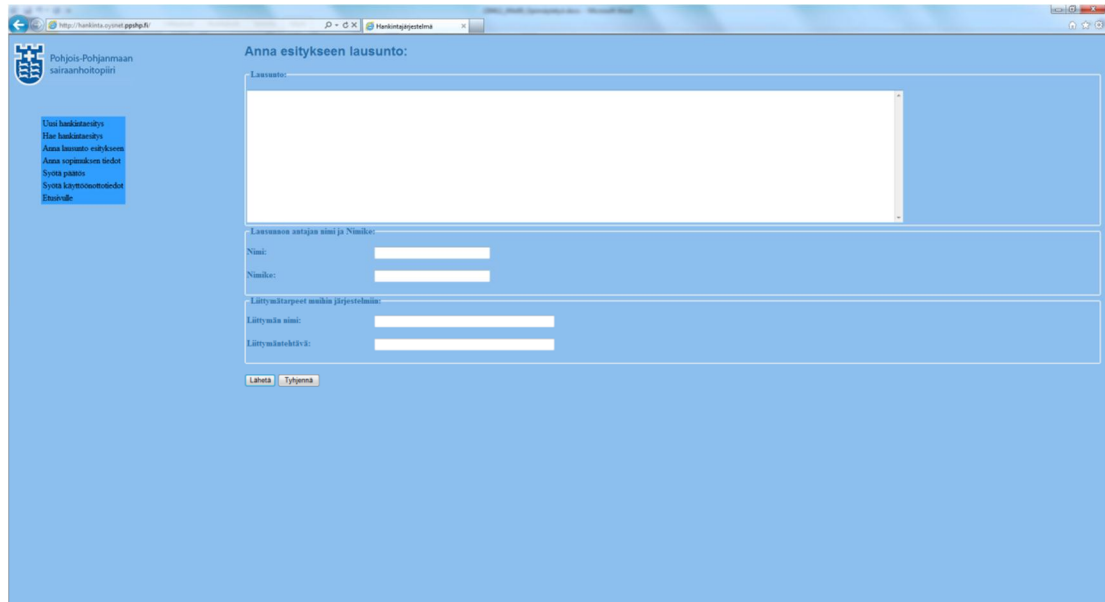
Kuva 9. Uusi hankintaesitys lomake

Lausunnon lisäys

Lausunnon lisäämiseksi hankintaesitykseen päävalikossa on öAnna lausunto esitykseenö-valinta, jonka valitsemalla avautuu hakulomake. Hakulomakkeella haetaan hankintaesitys johon lausuntoa ollaan antamassa, haku tapahtuu esitysnumerolla (Kuva 10).

Kuva 10. Hakulomake lausunnon antoa varten

Hakutuloksena saadaan taulukko, jossa on esityksen tietoja joiden avulla voidaan tarkistaa, että ollaan antamassa lausuntoa oikeaan esitykseen. Hakutuloksessa on linkki, jolla päästään lausuntolomakkeelle kirjoittamaan lausuntoa hankintaesitykseen (Kuva 11).

The image shows a web browser window with the URL 'http://hankinta.zymer.fi/psp/hp.fi'. The page title is 'Hankintajärjestelmä'. The main heading is 'Anna esityksen lausunto:'. Below this is a large text input field for the statement. To the left of the text field is a sidebar with a logo and a list of links: 'Uusi hankintatila', 'Hae hankintatila', 'Anna lausunto - vaikeus', 'Anna sopimuskäytön tiedot', 'Syötä päätös', 'Syötä käyttöönottotiedot', and 'Etsintä'. Below the text field are three input fields: 'Lausunnon antajan nimi ja Nimi:', 'Nimi:', 'Sukunimi:', and 'Lähtymätarpeet mukaan järjestelmään:'. Below these are two more input fields: 'Lähtymä nimi:' and 'Lähtymäteksti:'. At the bottom are two buttons: 'Läheta' and 'Tyhjennä'.

Kuva 11. Lausunnon syöttölomake

Lähetettyään tiedot palvelimelle, käyttäjä saa ilmoituksen onnistuneesta tietojen lisäyksestä tietokantaan tai mahdollisessa vikatilanteessa virheilmoituksen lisäyksen epäonnistumisesta.

Käyttöönottotietojen lisäys

Käyttöönottotietojen lisäämiseksi valikossa on valinta öSyötä käyttöönottotiedot. Valinnalla avautuu hakulomake. Hakulomakkeella haetaan hankintaesitys johon käyttöönottotietoja ollaan lisäämässä, haku tapahtuu esitysnumerolla. Haun tuloksena saadaan yksi tulosrivi, jota klikkaamalla avautuu käyttöönottotietolomake (Kuva 12).

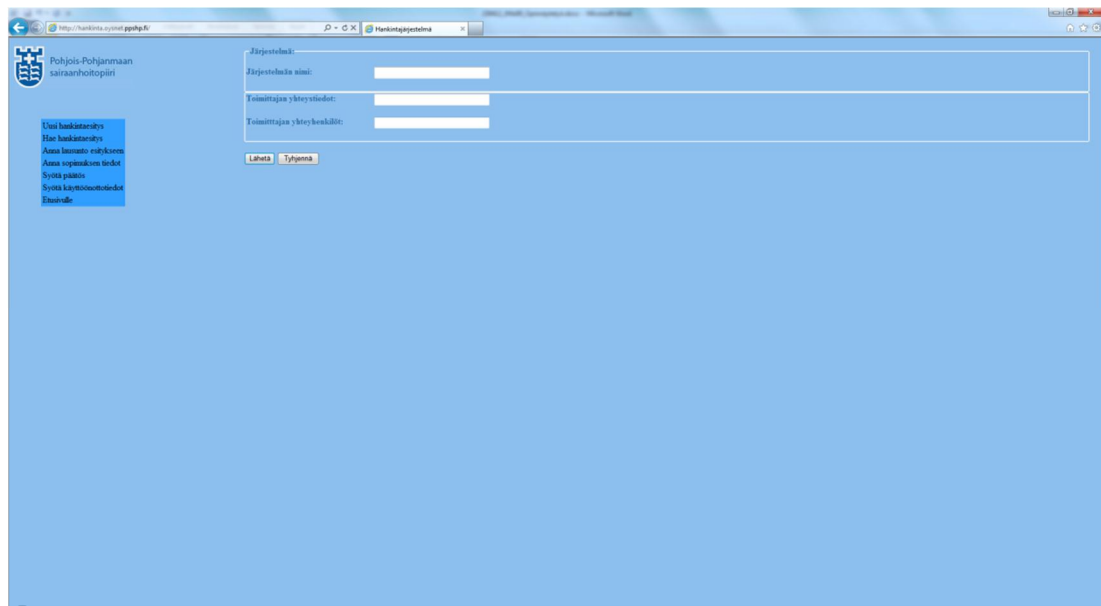
The screenshot shows a web browser window with the URL <http://hankinta.systest.psp.fi>. The page title is "Hankintajärjestelmä". The sidebar on the left contains the logo of Pohjois-Pohjanmaan sairaanhoitopiiri and a menu with the following items: "Uusi hankintasuoritus", "Hae hankintasuoritus", "Anna lausunto esitykseen", "Anna sopimuksen tiedot", "Syötä päätös", "Syötä käyttöönottotiedot", and "Etusivu". The main content area is titled "Käyttöönottoa edellyttävät tiedot:" and contains several input fields: "Eiikallista ja tarvittavat lisätietokentätiedot:" (a large text area), "Tietokanta:" (a text field), "Varmistukset:" (a text field), "Pääkkyttäjät (yhteensä enintään 2 henkilöä):" (a text field), "Pääkkyttäjät 1:" (a text field), "Pääkkyttäjät 2:" (a text field), "Ylläpitokäyttäjät:" (a text field), "Järjestelmän toimittajan ylläpiton varten tarvittavat lisätietokentätiedot:" (a text field), "Aiemmin käytetty:" (a text field), "Palvelu:" (a large text area), "Tietokanta:" (a text field), and "Tietokanta:" (a text field).

Kuva 12. Osa käyttöönottotietojen syöttölomakkeesta

Lähetettyään tiedot palvelimelle, käyttäjä saa ilmoituksen onnistuneesta tietojen lisäyksestä tietokantaan tai mahdollisessa vikatilanteessa virheilmoituksen lisäyksen epäonnistumisesta.

Hankintasopimuksen tietojen lisääminen

Hankintasopimuksen lisäämiseksi valikosta löytyy öAnna sopimuksen tiedotö valinta, jonka valitsemalla avautuu hakulomake. Hakulomakkeen avulla haetaan tietokannasta hankintaesitysnumeron perusteella se hankintaesitys, jonka sopimustietoja ollaan lisäämässä. Valitsemalla hakutulosten linkki, avautuu sopimustietojen syöttölomake (Kuva 13).

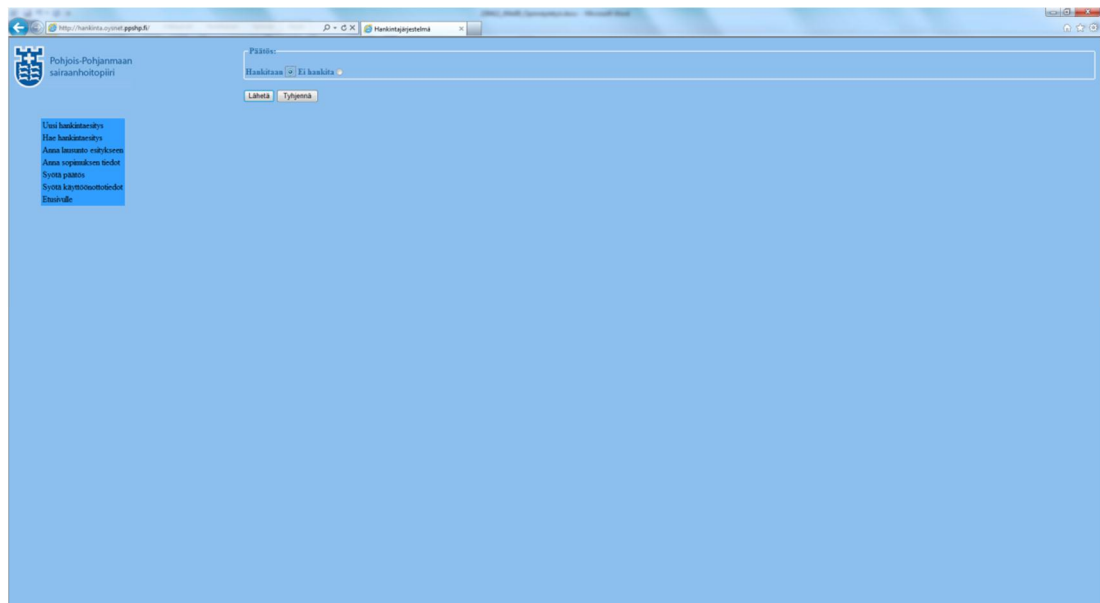


Kuva 13. Sopimustietojen syöttölomake

Lähetettyään tiedot palvelimelle, käyttäjä saa ilmoituksen onnistuneesta tietojen lisäyksestä tietokantaan tai mahdollisessa vikatilanteessa virheilmoituksen lisäyksen epäonnistumisesta.

Päätöksen kirjaaminen esitykseen

Päätöksen kirjaamiseksi valikossa on öSyötä päätösö valinta, jonka valitsemalla avautuu hakulomake. Hakulomakkeen avulla haetaan tietokannasta hankintaesitysnumeron perusteella, se hankintaesitys jonka päätös halutaan syöttää. Valitsemalla hakutuloksessa olevaa linkkiä avautuu päätöksen syöttölomake (Kuva 14).



Kuva 14. Päätöksen syöttölomake

Lähetettyään päätöksen palvelimelle, käyttäjä saa ilmoituksen onnistuneesta tiedon lisäyksestä tietokantaan tai mahdollisessa vikatilanteessa virheilmoituksen lisäyksen epäonnistumisesta.

Hankintaesitysten haku

Hankintaesitysten hakemiseksi valitaan valikosta öHae hankintaesitysö, valinta avaa uuden valikon josta voidaan valita hakuehdoksi esityksen numero, esityksen tekijä, esitetty tuote tai sitten voidaan listata kaikki tehdyt esitykset (Kuva 15).

Ehdotus numero	Ohjelma	Käyttötarkoitus	Yksikkö	Yhteyshenkilö	Ehdotuksen tekijä
1	MS Office 2010	laskutus	Tietohallinto	Matti Pohjola	Mari Hiltunen
2	MS Access	laskutus	KNF	Aleksi Alenka	Mari Hiltunen
3	CorelDraw	laskutus	Kirjasto	Jarmo Korpela	Jarmo Korpela
4	CorelPower	laskutus	Kirjasto	Jarmo Korpela	Jarmo Korpela
5	MS Share Point	laskutus	Tietohallinto	Pentti Korhonen	Mari Hiltunen
6	Windows 8	laskutus	Tietohallinto	Pentti Korhonen	Matti Pohjola
7	Windows 8.1	laskutus	Tietohallinto	Pentti Korhonen	Matti Pohjola
8	MS Office 2010 Pro	laskutus	KNF	Taina Kallio	Kalevi Kallio
9	MS Data	laskutus	Kirjasto	Kalle Korpela	Jarmo Kallio
10	PHP 5.3	laskutus	Tietohallinto	Lena Korpela	Ulla Korpela
11	Joomla!	laskutus	Huolto	Timo Tappanen	Jarmo Kallio
12	MS Paint	laskutus	Tietohallinto	Pentti Korhonen	Kalevi Kallio
13	Käsikirja	laskutus	KNF	Kalle Korpela	Ulla Korpela
14	Joomla-CMS	laskutus	Yhteistyö	Jarmo Kallio	Jarmo Kallio

Kuva 15. Hakutulostaulukko

Esityksiin pääsee tutustumaan tarkemmin tulostaulukon linkeistä. Jokaisesta ehdotuksesta löytyy tiedot liitteistä, lausunnoista, yms. syötetyistä tiedoista.

4.2. Sovelluksen testaus

Projektiin ei sisällynyt erillistä testausta lainkaan, vaan testaaminen tapahtui normaalin ohjelmakehityksen yhteydessä. Tämän projektin tuotoksena oli sovelluksen testiversio, jonka testauksen ja pilotoinnin tilaaja aloittaa omien aikataulujensa mukaisesti.

5. YHTEENVETO

Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa Pohjois-Pohjanmaan Sairaanhoidopiirille testiversio hankintasovelluksesta. Sovelluksen käyttöliittymän tuli olla www-selainpohjainen ja tiedon säilytyspaikkana käytettäisiin tilaajan MS SQL-palvelinta joka sijaitsee tilaajan sisäverkossa. Suunnittelu aloitettiin käymällä läpi nykyisen hankintaprosessin vaiheet. Hankintaprosessi sinällään oli toimiva ja soveltui sellaisenaan sovelluksen suunnittelun perustaksi.

Työ oli laajempi kokonaisuus kuin alun arvioinnissa näytti ja sen vuoksi loppuvaiheessa jouduttiin suorittamaan tarkempia rajoituksia toiminnallisuuksiin, jotta työ saataisiin valmiiksi viimeisen aikataulun mukaisesti. Ongelmana näin jälkikäteen voidaan nähdä myös JavaScript-kielen outous ja siihen tutustumisen vaatima aika, joka oli poissa varsinaisesta työn tekemisestä.

Jatkokehityksen kannalta sovellus on vaiheessa josta sen kehitystyötä on helppo jatkaa. Tietokanta on suunniteltu siten että sen mahdollinen laajentaminen onnistuu varsin helposti tauluja lisäämällä. Palvelinohjelmiston modulaarisuus mahdollista uusien moduulien helpon lisäämisen, käyttöliittymän ulkoasun muokkaus onnistuu CSS-tyylitiedostoja muokkaamalla ja toiminnallisuuksien muokkaus onnistuu erillisiä JavaScript-tiedostoja muokkaamalla.

Työn tulokseen olen, käytettävissä olleen työajan huomioon ottaen, tyytyväinen. Virhearvio työmäärissä heikensi työntuloksen tasoa ja aiheutti sovelluksen ominaisuuksien karsimista.

6. LÄHDELUETTELO

- /1/ Castro, Elizabeth, Kotisivut kuntoon- HTML, XHTML, CSS, 11. painos, Readme.fi, 2007.
- /2/ Huotari Jouni ja Hovi Ari, Tietokantojen suunnittelu, Jyväskylän Ammattikorkeakoulu,[www-dokumentti],
http://homes.jamk.fi/~huojo/opetus/IIZO3030/IIZO3030_03.pdf, 1.4.2012.
- /3/ Jyväskylän Yliopiston IT-tiedekunta ja avoin yliopisto, Relaatiotietokantojen peruskäsitteet [www-dokumentti],
[\[http://appro.mit.jyu.fi/doc/tiedonhallinta/tietokannat/index2.html\]](http://appro.mit.jyu.fi/doc/tiedonhallinta/tietokannat/index2.html), 3.3.2012.
- /4/ Kinnunen, Elina, Tietokannat, [www-dokumentti], [\[http://ku-tietotekniikka.wikispaces.com/Tietokannat\]](http://ku-tietotekniikka.wikispaces.com/Tietokannat), 22.2.2012.
- /5/ Mustaniemi Johanna, Käytettävyyden arviointimenetelmät, [www-dokumentti],
[\[https://jyx.jyu.fi/dspace/bitstream/handle/123456789/19970/Johanna.Mustaniemi.pdf?sequence=1\]](https://jyx.jyu.fi/dspace/bitstream/handle/123456789/19970/Johanna.Mustaniemi.pdf?sequence=1), 20.4.2012.
- /6/ Negrino Tom ja Smith Dori, JavaScript Tehokas hallinta1.painos, Readme.fi, 2007.
- /7/ Nielsen, Jacob, Jacob Nielsen Website, [www-dokumentti],[<http://www.useit.com/>], 15.4.2012.
- /8/ PHP.NET, History of PHP and Related Projects, [www-dokumentti],
[\[http://www.php.net/manual/en/history.php.php\]](http://www.php.net/manual/en/history.php.php), 23.3.2012.
- /9/ Salmela, Eila Helena, Tietokantojen historia ennen SQL:ää, [www-dokumentti],
[\[http://www.cs.helsinki.fi/u/kerola/tkhist/k2007/alustukset/tietokannat/TiKaHistoria.pdf\]](http://www.cs.helsinki.fi/u/kerola/tkhist/k2007/alustukset/tietokannat/TiKaHistoria.pdf) 4.3.2012.

7. LIITELUETTELO

Liite 1 Tietokantataulujen SQL-kielinen luontiscripti.

Liite 2 PHP-kielisen lomakkeen käsittely moduulin ohjelmalistaus.

Liite 3 CSS-tyylitiedoston listaus.

LIITE 1

```
CREATE TABLE hankinta.dbo.esitys
(
esitys_id int CONSTRAINT pk_esitys_pid PRIMARY KEY IDENTITY,
ohjelman_nimi varchar(50) NOT NULL,
ohjelman_kayttotarkoitus varchar(50) NOT NULL,
yksikon_nimi varchar(50) NOT NULL,
yksikon_nro varchar(50) NOT NULL,
yksikon_yhteyshenkilo varchar(50) NOT NULL,
yksikon_yhteyshenkilo_puh_nro varchar(50) NOT NULL,
yksikon_yhteyshenkilo_email varchar(50) NOT NULL,
ohjelman_hinta_arvio float NOT NULL,
Yksikon_budjetoima_hinta float NULL,
toiminnallisuuskuvauksen_tehy bit NOT NULL,
toiminnallisuuskuvauksen_liite_nimi varchar(30) NULL,
toiminnallisuuskuvauksen_liite_data varchar(max) NULL,
toiminnallisuuskuvauksen_liite_tyyppi varchar(30) NULL,
toiminnallisuuskuvauksen_liite_koko varchar(50) NULL,
kayttoymparistokuvaus_tehy bit NOT NULL,
kayttoymparistokuvaus_liite_nimi varchar(30) NULL,
kayttoymparistokuvaus_liite_data varchar(max) NULL,
kayttoymparistokuvaus_liite_tyyppi varchar(30) NULL,
kayttoymparistokuvaus_liite_koko varchar(50) NULL,
vaatimusmaarittely_tehy bit NOT NULL,
vaatimusmaarittely_liite_nimi varchar(30) NULL,
vaatimusmaarittely_liite_data varchar(max) NULL,
vaatimusmaarittely_liite_tyyppi varchar(30) NULL,
vaatimusmaarittely_liite_koko varchar(50) NULL,
pvm_klo varchar(30) NOT NULL,
```



```
esityksen_tekija_nimi varchar(50) NOT NULL,  
esityksen_tekija_nimike varchar(50) NULL,  
esityksen_tekija_email varchar(50) NOT NULL  
)
```

```
CREATE TABLE hankinta.dbo.lausunto  
(  
    lausunto_id int CONSTRAINT pk_lausunto_sid PRIMARY KEY IDENTITY,  
    lausunto varchar(max) NOT NULL,  
    pvm varchar(30) NOT NULL,  
    lausunnon_antaja_nimi varchar(50) NOT NULL,  
    lausunnon_antaja_nimike varchar(30) NOT NULL,  
    esitys_id int CONSTRAINT fk_lausunto_pid FOREIGN KEY REFERENCES  
    hankinta.dbo.esitys(esitys_id) NOT NULL  
)
```

```
CREATE TABLE hankinta.dbo.liianta  
(  
    liianta_id int CONSTRAINT pk_liianta_sid PRIMARY KEY IDENTITY,  
    liianta_muut_jarjestelmat_nimi varchar(50) NULL,  
    liianta_muut_jarjestelmat_tehtava varchar(max) NULL,  
    lausunto_id int CONSTRAINT fk_liianta_pid FOREIGN KEY REFERENCES  
    hankinta.dbo.lausunto(lausunto_id) NOT NULL  
)
```

```
CREATE TABLE hankinta.dbo.paatos  
(  
    paatos_id int CONSTRAINT pk_paatos_sid PRIMARY KEY IDENTITY,  
    paatos_kylla_ei bit NOT NULL,  
    pvm varchar(30) NOT NULL,  
    esitys_id int CONSTRAINT fk_paatos_pid FOREIGN KEY REFERENCES
```

```
hankinta.dbo.esitys(esitys_id) NOT NULL
)
```

```
CREATE TABLE hankinta.dbo.sopimus
(
sopimus_id int CONSTRAINT pk_sopimus_sid PRIMARY KEY IDENTITY,
ohjelma varchar(50) NOT NULL,
ohjelma_toimittaja_yhteys varchar(50) NOT NULL,
ohjelma_toimittaja_yhteys_hlo varchar(50) NOT NULL,
esitys_id int CONSTRAINT fk_sopimus_pid FOREIGN KEY REFERENCES
esitys(esitys_id) NOT NULL
)
```

```
CREATE TABLE hankinta.dbo.kayttoonotto
(
kayttoonotto_id int CONSTRAINT pk_kayttoonotto_sid PRIMARY KEY IDENTITY,
eh_tly varchar(max) NOT NULL,
tietokanta varchar(50) NULL,
varmuuskopio varchar(50) NOT NULL,
yllapitaja_1 varchar(50) NOT NULL,
yllapitaja_2 varchar(50) NOT NULL,
yllapito_sopimus varchar(50) NOT NULL,
yllapito_tietoliikenne varchar(max) NULL,
palvelin varchar(max),
tietokantaa varchar(max),
wks varchar(max),
liittymat varchar(max),
tpalvelut varchar(max),
tjarjestelmat varchar(max),
```

```
hankinta_tsto_nimi varchar(50) NOT NULL,  
hankinta_tsto_nimike varchar(30) NOT NULL,  
hankinta_tsto_pvm varchar(30) NOT NULL,  
th_nimi varchar(50) NOT NULL,  
th_nimike varchar(30) NOT NULL,  
th_pvm varchar(30) NOT NULL,  
esitys_id int CONSTRAINT fk_kayttoonotto_pid FOREIGN KEY REFERENCES  
esitys(esitys_id) NOT NULL,  
)
```

LIITE 2

PHP ohjelmamoduuli uuden hankintaesityksen luomiseksi

```
<?php
```

```
//*****Lisataan uusi hankintaesitys tietokantaan*****
```

```
function lisaa_esitys($connect){
```

```
    // $day = date("d.m.Y H:i");
```

```
    $ohjelma = addslashes( $_POST['nimi']);
```

```
    $ktarkoitus = addslashes( $_POST['ktarkoitus']);
```

```
    $yksikko = addslashes( $_POST['yksikko']);
```

```
    $yksikkonro = addslashes( $_POST['yksikko_nro']);
```

```
    $yhenkilo = addslashes( $_POST['yhteysh']);
```

```
    $yhenkilopuhnro = addslashes( $_POST['yhteysh_nro']);
```

```
    $yhenkiloemail = addslashes($_POST['yemail']);
```

```
    $hinta_arvio = addslashes( $_POST['hinta']);
```

```
    $budjetti = addslashes( $_POST['budjetti']);
```

```
    $tkuvaus = addslashes($_POST['tkuvaus']);
```

```
    $kykuvaus = addslashes( $_POST['kykuvaus']);
```

```
    $vmaarittely = addslashes($_POST['vmaarittely']);
```

```
    $esityksen_tekija = addslashes($_POST['esityksen_tekija']);
```

```
    $tnimike = addslashes($_POST['tekijan_nimike']);
```

```
    $temail = addslashes($_POST['tekijan_email']);
```

```
    $tmp_dir_move = "data/";
```

```
//Tarkistetaan onko toiminnallisuuskuvaus dokumenttia liitteenä
```

```
    if ($tkuvaus == "true"){ //tarkistetaan onko toiminnallisuuskuvausta tehty
```

```
$tmp_file = $_FILES['tkuvaus_liite']['tmp_name'];
$file_name_tkuvaus = $_FILES['tkuvaus_liite']['name'];

$file_path_tkuvaus = $tmp_dir_move.$file_name_tkuvaus;
$file_type_tkuvaus = $_FILES['tkuvaus_liite']['type'];
$file_size_tkuvaus = $_FILES['tkuvaus_liite']['size'];

$result_tk = move_uploaded_file($tmp_file,$file_path_tkuvaus);

$fh_tkuvaus = fopen($file_path_tkuvaus, "rb");

$data_tkuvaus = addslashes(fread($fh_tkuvaus,filesize($file_path_tkuvaus)));
$data_tkuvaus = bin2hex($data_tkuvaus);
$data_tkuvaus = "0x".$data_tkuvaus;

fclose($fh_tkuvaus);
unlink($file_path_tkuvaus);
}
else {
    $data_tkuvaus = NULL;
    $file_type_tkuvaus = NULL;
    $file_name_tkuvaus = NULL;
    $file_size_tkuvaus = NULL;
}

//Tarkistetaan onko käyttöympäristökuvaus dokumenttia liitteenä
if ($kykuvaus == "true"){

    $tmp_file_ky = $_FILES['kykuvaus_liite']['tmp_name'];
    $file_name_kykuvaus = $_FILES['kykuvaus_liite']['name'];
```

```
$file_path_kykuvaus = $tmp_dir_move.$file_name_tkuvaus;
$file_type_kykuvaus = $_FILES['kykuvaus_liite']['type'];
$file_size_kykuvaus = $_FILES['kykuvaus_liite']['size'];

$result_ky = move_uploaded_file($tmp_file_ky,$file_path_kykuvaus);

$fh_kykuvaus = fopen($file_path_kykuvaus, "rb");

$data_kykuvaus = addslashes(fread($fh_kykuvaus,filesize($file_path_kykuvaus)));
$data_kykuvaus = bin2hex($data_kykuvaus);
$data_kykuvaus = "0x".$data_kykuvaus;

fclose($fh_kykuvaus);
unlink($file_path_kykuvaus);
}
else {
    $data_kykuvaus = NULL;
    $file_type_kykuvaus = NULL;
    $file_name_kykuvaus = NULL;
    $file_size_kykuvaus = NULL;
}

//Tarkistetaan onko vaatimusmäärittely dokumenttia liitteenä
if ($vmaarittely == "true"){

    $tmp_file_vm = $_FILES['vmaarittely_liite']['tmp_name'];
    $file_name_vm = $_FILES['vmaarittely_liite']['name'];

    $file_path_vm = $tmp_dir_move.$file_name_vm;
    $file_type_vm = $_FILES['vmaarittely_liite']['type'];
    $file_size_vm = $_FILES['vmaarittely_liite']['size'];
```

```
$result_vm = move_uploaded_file($tmp_file_vm,$file_path_vm);

    $fh_vmkuvaus = fopen($file_path_vm, "rb");
    $data_vmkuvaus = addslashes(fread($fh_vmkuvaus,filesize($file_path_vm)));
    $data_vmkuvaus = bin2hex($data_vmkuvaus);
    $data_vm = "0x".$data_vmkuvaus;

    fclose($fh_vmkuvaus);
    unlink($file_path_vm);
}
else
{

    $data_vm = NULL;
    $file_type_vm = NULL;
    $file_name_vm = NULL;
    $file_size_vm = NULL;

}

$day_time = date("d.m.Y H:i");

$mssql = ("INSERT INTO dbo.esitys
            (ohjelman_nimi,
            ohjelman_kayttotarkoitus,
            yksikon_nimi,
            yksikon_nro,
            yksikon_yhteyshenkilo,
            yksikon_yhteyshenkilo_puh_nro,
            yksikon_yhteyshenkilo_email,
            ohjelman_hinta_arvio,
            yksikon_budjetoima_hinta,
```

```
toiminnallisuuskuvaus_tehty,  
toiminnallisuuskuvaus_liite_nimi,  
toiminnallisuuskuvaus_liite_data,  
toiminnallisuuskuvaus_liite_tyyppi,  
toiminnallisuuskuvaus_liite_koko,  
kayttoymparistokuvaus_tehty,  
kayttoymparistokuvaus_liite_nimi,  
kayttoymparistokuvaus_liite_data,  
kayttoymparistokuvaus_liite_tyyppi,  
kayttoymparistokuvaus_liite_koko,  
vaatimusmaarittely_tehty,  
vaatimusmaarittely_liite_nimi,  
vaatimusmaarittely_liite_data,  
vaatimusmaarittely_liite_tyyppi,  
vaatimusmaarittely_liite_koko,  
pvm_klo,  
esityksen_tekija_nimi,  
esityksen_tekija_nimike,  
esityksen_tekija_email)  
VALUES ('$ohjelma',  
        '$ktarkoitus',  
        '$yksikko',  
        '$yksikkonro',  
        '$yhenkilo',  
        '$yhenkilopuhnro',  
        '$yhenkiloemail',  
        '$hinta_arvio',  
        '$budjetti',  
        '$tkuvaus',  
        '$file_name_tkuvaus',  
        '$data_tkuvaus',
```



```
'$file_type_tkuvaus',  
'$file_size_tkuvaus',  
'$kykuvaus',  
'$file_name_kykuvaus',  
'$data_kykuvaus',  
'$file_type_kykuvaus',  
'$file_size_kykuvaus',  
'$vmaarittely',  
'$file_name_vm',  
'$data_vm',  
'$file_type_vm',  
'$file_size_vm',  
'$day_time',  
'$esityksen_tekija',  
'$tnimike',  
'$temail')");
```

```
$query = mssql_query($mssql); //kirjoitetaan lomakkeesta saadut datat tietokantaan
```

```
if (!$query) // tarkistetaan onnistuiko tietokantaan kirjoitus  
{  
    echo "<br>";  
    echo "<br>";  
    echo "<br>";  
    echo "<center>";  
    echo "Tietoja ei tallennettu<br>";  
    echo 'MSSQL error: '. mssql_get_last_message();  
    echo "</center>";  
}
```


LIITE 3

Listaus hankintasovelluksessa käytetystä tyylitiedostosta.

```
body {  
background-color:#8CBFEE;  
color:#276CA7;  
}  
div#body_text {  
background-color:#8CBFEE;  
color:#276CA7;  
right:40px;  
font-family: sans-serif;  
font-size: 24px;  
font-weight: bold;  
margin-left: 20px;  
}  
FORM {  
margin-left: 20px;  
font-weight: bold;  
}  
  
div#top {  
font-size:56px;  
background-color:#8CBFEE;  
color:#276CA7;  
}  
div#main {  
font-size: 48px;  
font-weight: bold;  
text-align: center;  
vertical-align: baseline;  
}  
div #form {  
caption-side: inherit;  
  
}  
  
H1 {  
background-color:#8CBFEE;  
color:#276CA7;  
}  
H2 {  
background-color:#8CBFEE;  
color:#276CA7;  
}  
table, tr, td {
```

```
background-color:#015CA5;
color:#276CA7;
border-width:2px;
border-color:#8CBFEE;
margin-left: 50px;
}
table, tr, td, A:link {
text-decoration: none;
color:#000000;
}
table, tr, td, A:visited {
text-decoration: none;
color: #000000;
}
table, tr, td, A:active {
text-decoration: none;
color: #000000;
}
table, tr, td, A:hover {
text-decoration: none;
color: #FFFFFF;
}
legend {
background-color:#8CBFEE;
color:#276CA7;
}
fieldset {
margin-left:auto;

}
.vasen {
width: 10%;
float: left;
}
div #menu_img{
width: 10%;
float: left;
}

input.invalid{
background-color:#F00;
border:2px red inset;
}
```